TABLE NO | | | |

STUDENT ID NO | | | | | | | | | | | |

MULTIMEDIA UNIVERSITY

SUBJECT CODE _____

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 2, 2018/2019

TLD7011 – LOW LEVEL DESIGN OF SOFTWARE

(All sections / Groups)

25 JANUARY 2018
2:30 pm – 4:30 pm
(2 Hours)

Question	Mark
A	
B	
C	
D	
Total	

Examiner 1 Signature: _____

Examiner 2 Signature: _____

Examiner 3 Signature: _____

INSTRUCTIONS TO STUDENTS

1. This question paper consists of 6 printed pages (including cover page) with 4 Sections only.

2. Attempt **ALL** questions in **SECTION A, SECTION B, SECTION C and SECTION D**. The distribution of the marks for each question is given.

3. Please write all your answers **CLEARLY** in the answer booklet provided.

YOU MUST ANSWER EVERY QUESTION. EACH SECTION CARRIES THE SAME DISTRIBUTION OF MARKS.

SECTION A

A-1. Figure 1 shows a task dependency graph. Assuming that each task requires 1 unit of time, calculate the following metrics with workings.

[1 + 2 + 2 MARKS]

I. Critical path length
II. Total amount of work
III. Average degree of concurrency

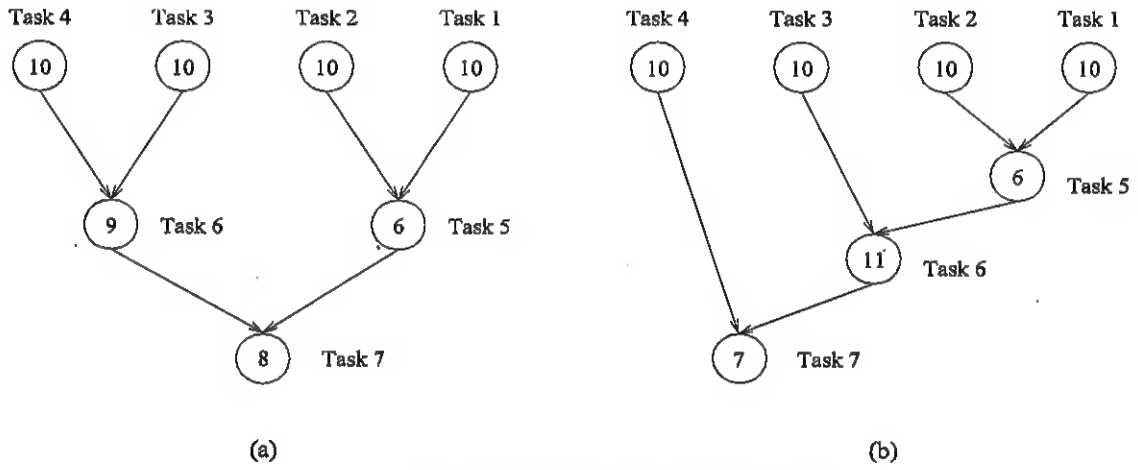

Fig. 1 Task Dependency Diagram

A-2. Figure 2 shows the process of counting the number of instances of a given item-sets in a database of transactions. The output as shown in Figure 2, would be the item-set frequencies. Draw a single diagram that demonstrates how the output data decomposition is performed in this scenario.

[2 MARKS]

Continued...

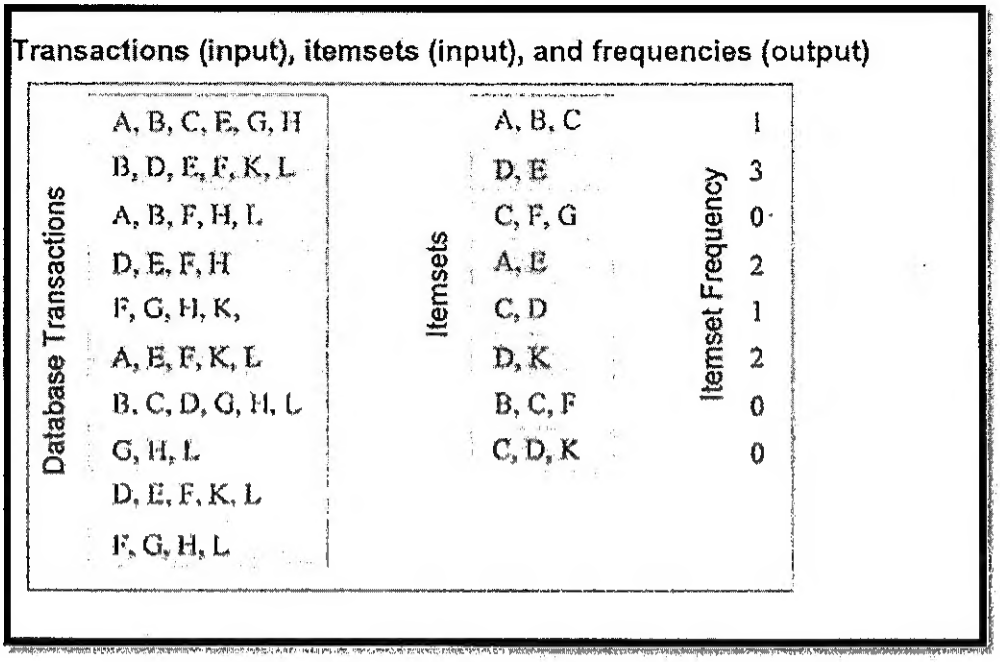

Fig. 2 Transaction, item-sets and frequencies of a given database

A-3. Draw and explain how a typical SIMD (Single Instruction, Multiple Data) architecture differ from a typical MIMD (Multiple Instruction, Multiple Data) architecture.

[3 MARKS]

SECTION B

B-1. Consider the following snippet of code. We want to be able to **construct** an object representing a cat knowing:

- only her age,
- or only her age and weight,
- or only her age, weight and length,
- or only her age, weight, length and id
- or only her age, weight, length, id and name

Identify one shortcoming of this design, and subsequently identify a suitable software design pattern which can be used to improve the design of the code, and give a reason why this software design pattern improves the design.

[3 MARKS]

Continued...

```
public class Cat {
        private int age;
        private int weigth;
        private int length;
        private int id;
        private String name;

        //////////////Here comes the telescopic constructors
        public Cat() {
                //some stuff
        }

        public Cat(int age) {
                this();//we're using the previous constructor
                this.age = age;
        }

        public Cat(int age, int weigth) {
                this(age);//we're using the previous constructor
                this.weigth = weigth;
        }

        public Cat(int age, int weigth, int length) {
                this(age, weigth);
                //we're using the previous constructor
                this.length= length;
        }

        public Cat(int age, int weigth, int length, int id) {
                this(age, weigth, length);
                //we're using the previous constructor
                this.id = id;
        }

        public Cat(int age, int weigth, int length, int id, String
name) {
                this(age, weigth, length, id);
                //we're using the previous constructor
                this.name = name;
        }

}
```

B-2. Draw the generic class diagram for the design pattern, which you have identified in Question B-1.

[3 MARKS]

B-3. Write out the code (pseudo code acceptable in lieu of actual code) that implements the design pattern that you have identified.

[4 MARKS]

Continued...

SECTION C

C-1. Describe the FOUR activities that you do in refactoring.

[4 MARKS]

C-2. For each of the following codes, re-factor using the given method, pseudocode is acceptable. State the motivation to use the particular refactoring technique, and any special considerations that needs to be addressed.

[6 MARKS]

I. Encapsulate Field

```
1: public class Person
2: {
3:     public string CommonName;
4:     public string Course;
5:     public int Age;
6: }
```

II. Extract Hierarchy

```
1: public class Sheep
2: {
3:     public void eatFood()
4:     {
5:         // eat some food
6:     }
7:
8:     public void brushHair()
9:     {
10:        // brushing hair
11:    }
12: }
```

Continued...

III. Extract Method

```
 1: public class CafeReceipt
 2: {
 3:     private IList<decimal> Discounts { get; set; }
 4:     private IList<decimal> ItemTotals { get; set; }
 5:
 6:     public decimal CalculateGrandTotal()
 7:     {
 8:         decimal subTotal = 0m;
 9:         foreach (decimal itemTotal in ItemTotals)
10:             subTotal += itemTotal;
11:
12:         if (Discounts.Count > 0)
13:         {
14:             foreach (decimal discount in Discounts)
15:                 subTotal -= discount;
16:         }
17:
18:         decimal tax = subTotal * 0.060m;
19:
20:         subTotal += tax;
21:
22:         return subTotal;
23:     }
24: }
```

SECTION D

Pair programming involves the practice of developing software in groups of two people, one at the keyboard and one standing by, speaking out their thought processes to each other. XP (Extreme Programming) prescribes pair programming as the standard practice. Andy and Bob works as a team when they are developing a software solution. Bob is the junior developer so he codes, while Andy observes. Andy's role is to point out mistakes and to plan ahead. Subsequently, their work will be presented together during a software design review. In the review, Andy and Bob play the role of the author.

D-1. State FOUR benefits of Andy and Bob's way of working.

[4 MARKS]

D-2. List FOUR methods of software design review.

[2 MARKS]

D-3. Explain FOUR advantages of using the agile methodology as compared to using a traditional waterfall development methodology.

[4 MARKS]

END of Paper